Claims:

1. An apparatus (299) for processing a stream of fixed-length packets received as digitally encoded signals and having multiple packet types, each packet including a header portion and a data portion, the header portion containing a sync byte, the apparatus comprising:

a Null-Packet Detector (250) for detecting whether a received packet is a null-packet and for identifying the location of the sync-byte of a detected null-packet.

2. The apparatus of claim 1, wherein the Null-Packet Detector (250) further generates a Null_flag signal to indicate whether a received packet is a null-packet and generates a Null_sync signal to indicate the location of the sync-byte of a detected null-packet.

3. The apparatus of claim 2, further comprising a circuit (240) adapted to insert a predetermined sync-byte value into the sync-byte position indicated by the Null_sync signal.

4. The apparatus of claim 2, further comprising a filter (255) adapted to filter the Null_flag signal and adapted to generate a Null_lock signal a first value of which indicates that the stream contains a plurality of null packets.

5. The apparatus of claim 4 wherein the filter (255) implements hysteresis thresholding.

6. The apparatus of claim 4 wherein the filter (255) is implemented by a finite state machine.

7. The apparatus of claim 4, wherein the first value of Null_lock signal output by the filter (255) indicates that the stream contains a first threshold number of null-packets (Lock_In_thresh ) within a first number of consecutive packets.

8. The apparatus of claim 7, wherein at least one of the first threshold number and the first number of consecutive packets is programmable.

9. The apparatus of claim 7, wherein a second value of the Null_lock signal output by the filter (255) indicates that the stream contains a second threshold number (Lock_Out_thresh) of packets that are not null packets, within a second number of consecutive packets.

10. The apparatus of claim 8, wherein at least one of the first threshold number, and the second threshold number is programmable.

11. The apparatus of claim 2, wherein the Null-Packet Detector (250) determines whether a received packet is a null-packet by comparing contents of the header portion of the received packet with a first predetermined value.

12. The apparatus of claim 11, wherein the Null-Packet Detector (250) determines whether a received packet is a null-packet by further comparing contents of the data portion of the received packet with a second predetermined value.

13. An apparatus (299) for processing a stream of fixed-length packets received as digitally encoded signals and having multiple packet types, each packet including a header portion and a data portion, the header portion containing a checksum-encoded sync byte, the apparatus comprising:

a Syndrome Detector (210) for detecting the checksum-encoded sync-byte and for generating a Sync_flag signal to indicate the location of the checksum-encoded sync-byte;

a Null-Packet Detector (250) adapted to detect whether a received packet is a null-packet, and adapted to identify the location of the sync-byte of a detected null-packet; and

an MPEG Sync-Byte Re-insertion circuit (240) for inserting a predetermined value into the sync-byte location indicated by an MPEG synchronization signal.

14. The apparatus of claim 13, wherein the Null-Packet Detector (250) is further adapted to output a Null_sync signal to indicate the location of the sync-byte of a detected null-packet.

15. The apparatus of claim 14, further comprising:

a multiplexor, wherein the Sync_flag output of the Syndrome Detector and the Null_sync output of the Null-Packet Detector are multiplexed and are alternatively output by the multiplexor, to be used by the MPEG Sync-Byte Re-insertion circuit, according to whether null packets have been detected.

16. The apparatus of claim 15, further comprising a decisional logic circuit (260) operatively connected to the multiplexor and adapted to control the multiplexor so that when the Null-Packet Detector detects null packets, the Null_sync output of the Null Packet Detector is output by the multiplexor to be used as the MPEG synchronization signal by the MPEG Sync-Byte Re-insertion circuit.

17. The apparatus of claim 14 adapted so that when null packets are detected, the Null_sync output of the Null Packet detector is used as the MPEG synchronization signal used by the MPEG Sync Re-insertion circuit.

18. The apparatus of claim 17, wherein when null packets are not detected, the Null_sync output of the Null Packet detector is not used as the MPEG synchronization signal used by the MPEG Sync Re-insertion circuit.

19. The apparatus of claim 18, wherein when null packets are not detected, the Sync_flag output by the Syndrome Detector is used as the MPEG synchronization signal used by the MPEG Sync Re-insertion circuit.

20. A method for processing a stream of fixed length packets each packet containing a checksum-encoded sync-byte, the stream including a plurality of packets that each contain a first fixed bit pattern in the header portion of each packet, the method comprising:

performing a first detection step of decoding the checksum in the stream to detect a checksum-encoded sync byte position candidate in the current one of the fixed length packets; and

performing a second detection step to detect the first fixed bit pattern in the header portion of the current one of the fixed length packets;

if the first fixed bit pattern is detected in the stream of fixed length packets, then identifying the sync-byte position of the sync-byte of each of the fixed length packets based upon the detection of the first fixed bit pattern;

inserting a predetermined sync-byte value into the identified sync-byte position.

21. The method of claim 20, wherein the second detection step is performed only if a checksum-encoded sync byte position candidate is detected in the first detection step.

22. The method of claim 20, wherein while the first fixed bit pattern is not detected in the stream of fixed length packets, then identifying the sync-byte position of the sync-byte of each of the fixed length packets in the stream based upon the result of the first detection step;

and inserting the predetermined sync-byte value into the sync-byte position candidate based upon the result of the first detection step.

23. The method of claim 20, wherein if a checksum-encoded sync byte candidate is detected in the first detection step, but the first fixed bit pattern is not detected by the second detection step, then inserting a predetermined sync-byte value into the sync-byte position candidate based upon the result of the first detection step.

24. The method of claim 20, wherein the second detection step is performed even if a checksum-encoded sync byte candidate is not detected in the first detection step.

25. The method of claim 20, wherein the first fixed bit pattern is a predetermined bit pattern in the header portion of each packet within the plurality of packets.

26. The method of claim 25, wherein said header portions comprise transport headers of an MPEG-2 Transport Stream.

27. The method of claim 20, wherein the first fixed bit pattern is a predetermined pattern that includes at least one of the following MPEG-2 transport stream link header field values: payload_unit_start_indicator = '0', PID=0x1FFF, transport scrambling control = '00', and adaptation field = '01'.

28. The method of claim 20, wherein each of the plurality of packets is an MPEG-2 null-packet.

29. The method of claim 20, wherein each packet in the plurality of packets that each contain a first fixed bit pattern in a header portion of each packet further contains a second fixed bit pattern within the data portion of each packet; and further comprising:

performing a third detection step to detect the second fixed bit pattern in the plurality of packets within the stream of fixed length packets, and if the second fixed bit pattern is detected in the stream of fixed length packets, then inserting the predetermined sync-byte value into the sync-byte position based upon the result of the third detection step.

30. The method of claim 28, wherein the third detection step is performed only if a checksum-encoded sync byte position candidate is detected in the first detection step.

31. The method of claim 28, wherein if neither of the first and second detection steps has identified a sync byte position, then no predetermined sync-byte value is inserted in the stream of fixed length packets.

32. The method of claim 28, wherein the third detection step is performed only if the first fixed bit pattern is detected in the stream of fixed length packets in the second detection step.

33. The method of claim 28, wherein if the second fixed bit pattern is not detected in the stream of fixed length packets, then identifying the sync-byte position of the sync-byte of each of the fixed length packets based upon the result of the first detection step.

34. A method for processing a stream of fixed length packets, each packet including a checksum-encoded sync-byte, the stream including a plurality of packets that each contain a first data pattern in a PID portion, the method comprising:

decoding the checksum in a preceding one of the fixed length packets to detect a checksum-encoded sync byte candidate in a current one of the fixed length packets; and

if a checksum-encoded sync byte candidate is detected in the decoding step, then searching for the first data pattern in the PID portion of the current one of the fixed length packets.

35. An apparatus for processing a stream of fixed length packets, each packet including a checksum-encoded sync-byte, the stream including a plurality of packets that each contain a first data pattern in a PID portion, the apparatus comprising:

means for decoding the checksum in a preceding one of the fixed length packets to detect a checksum-encoded sync byte candidate in a current one of the fixed length packets; and

means for searching for the first data pattern in the PID portion of the current one of the fixed length packets when a checksum-encoded sync byte candidate is detected in the decoding step.

36. A computer program product for a set-top-box that comprises a set of instructions, which, when loaded into the set-top-box, causes the set-top-box to carry out the method, for processing a stream of fixed length packets, claimed in claim 20.

37. A computer program product for a television set that comprises a set of instructions, which, when loaded into the television set, causes the television set to carry out the method, for processing a stream of fixed length packets, claimed in claim 20.